

Rancang Bangun *Inertial Measurement Unit* Untuk *Unmanned Aerial Vehicles* “Quadrotor”

Muhammad Alfiansyah, Rudy Dikairono dan Pujiono.

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: rudydikairono@ee.its.ac.id

Abstrak — Teknologi robot udara quadrotor semakin berkembang pesat. Salah satu bagian yang marak dikembangkan adalah sensor orientasinya atau yang umum disebut sebagai *Inertial Measurement Unit (IMU)*. Permasalahan pada *IMU* yang umum terjadi antara lain adalah ketidakmampuan *processing unit* untuk mengolah data dengan cepat (*output data rate* rendah), beban komputasi yang tinggi (algoritma penggabungan data sensor yang berat), luaran *Accelerometer* ber-noise tinggi yang umumnya berasal dari getaran body UAV dan luaran *Gyroscope* yang mengalami drift. Pada tugas akhir ini diciptakan sebuah *IMU* menggunakan Mikrokontroler *STM32F4* sebagai pemroses data dan metode determinasi orientasi dengan representasi *Direction Cosine Matrices* sebagai algoritma penyatuan data dan penentu luaran orientasi. Dari hasil pengujian menggunakan gimbal elektronik, dapat dicapai *RMS Error* statis dibawah 2 derajat dan *RMS Error* dinamis dibawah 3.4 derajat.

Kata Kunci—*Inertial Measurement Unit, Sensor Fusion, Direction cosine Matrices.*

I. PENDAHULUAN

QUADROTOR merupakan salah satu dari sekian banyak *Unmanned Aerial Vehicles (UAVs)* yang memiliki keunggulan struktur mekanik sederhana. Memiliki mekanik yang sederhana mengharuskan quadrotor untuk memiliki kontroler yang handal. Salah satu faktor penting dalam sebuah kontroler adalah kualitas dari masukannya. *IMU* atau yang berkepanjangan *Inertial Measurement Unit* merupakan sebuah bagian pada quadrotor yang bertugas memberikan data berupa data sudut orientasi. Untuk menjamin kehandalan dari kontroler quadrotor maka sebuah *IMU* wajib memberikan data sudut orientasi secara akurat. Untuk menghasilkan data yang akurat terdapat banyak permasalahan yang hingga kini masih menjadi topik pembicaraan yang hangat dalam pengembangan *IMU*. Dalam menghasilkan luaran sudut orientasi terhadap Bumi sebuah *IMU* minimnya hanya membutuhkan sebuah *Accelerometer*, namun dalam aplikasi kontrol navigasi hal ini akan mendatangkan masalah karena sensor ini sangat rentan terhadap noise getaran, padahal sebuah quadrotor terdiri atas minimal 4 buah motor yang bergetar kuat. Dalam berbagai aplikasi yang memiliki sifat dinamis, sebuah *gyroscope* dapat digunakan untuk menghasilkan orientasi dengan mendeteksi kecepatan sudut dimana sensor ini ditempatkan. Keunggulan *gyroscope* adalah tahan terhadap gangguan getaran, namun juga memiliki

kelemahan antara lain karena ada perubahan *bias* kondisi nolnya terhadap waktu yang mengakibatkan muncul *drift* saat mengintegrasikan luarannya menjadi sudut. Oleh sebab itu luaran *gyroscope* hanya *reliable* pada jangka waktu yang sebentar saja. Sudah menjadi metode umum dua sensor ini kerap digunakan bersama dalam satu sistem untuk melengkapi satu sama lain dalam menghasilkan luaran sudut orientasi. Permasalahan selanjutnya adalah berbagai algoritma yang sudah ada untuk menghasilkan luaran sudut dinilai terlalu berat dan rumit. Menurut Sebastian O.H. Madgwick dalam papernya, algoritma seperti kalman filter membutuhkan kemampuan komputasi yang tinggi serta sampling rate yang besar [1], dimana dua hal ini adalah harga yang sangat mahal untuk sebuah sistem *embedded*. Ia sendiri mengajukan penggunaan metode *Steepest gradien decent* dengan representasi *quaternion*. Metode yang diajukannya ini dikenal sebagai metode yang mampu menghasilkan error lebih kecil dari pada *IMU* yang menggunakan kalman filter dan dengan sampling rate yang tidak terlalu besar. Sebuah algoritma lain yang banyak digunakan penghobi quadrotor adalah algoritma *DCM (Direction Cosine Matrices)* karya William Premerlani dalam draft papernya. *DCM* sendiri sebenarnya bukanlah sebuah algoritma *sensor fusion* namun sebuah matriks rotasi yang melambangkan semua sudut antar 2 buah koordinat frame. Dari berbagai jenis metode menyatukan data tersebut dipilih sebuah metode buatan Sergiybaluta seperti yang terpapar dalam artikelnya [2]. Metode ini menggunakan *DCM* sebagai representasi sudut dan berbagai pendekatan matematis sederhana dalam meng-update data orientasi. Diharapkan dengan kesederhanaan algoritma ini dapat menyelesaikan permasalahan atas penggunaan algoritma lain yang tergolong berat seperti kalman filter. Pada Tugas akhir ini metode tersebut dikembangkan dengan diaplikasikan pada 3 buah sensor yakni *Accelerometer* 3 aksis, *Gyroscope* 3 aksis dan *Magnetometer* tiga aksis. Untuk pemenuhan kebutuhan kontrol navigasi, luaran utama dari sistem *IMU* ini adalah sudut global dengan urutan rotasi *roll*, *pitch* dan *yaw*. Data orientasi kemudian diuji dengan menggunakan gimbal elektronik dan pengamatan busur derajat. Pada bagian akhir akan diujikan dengan menempatkan *IMU* pada *Quadrotor* yang di hidupkan namun tidak diterbangkan untuk menguji kehandalan *IMU* hasil rancangan dengan kondisi lingkungan quadrotor.

II. PERANCANGAN SISTEM

Diagram blok perancangan sistem *IMU* dapat diamati pada Gambar 1. *Accelerometer* 3 aksis digunakan untuk mendeteksi vektor gravitasi Bumi, *Magnetometer* 3 aksis digunakan untuk mendeteksi vektor arah Utara, kedua duanya dilihat dari sudut pandang koordinat lokal. *Gyroscope* 3 aksis digunakan untuk mendapatkan kecepatan sudut yang digunakan untuk meng-



Gambar. 1. Diagram Blok Sistem *Inertial Measurement Unit*.

update posisi vektor gravitasi dan utara, kemudian semua masukkan tersebut diolah dalam Algoritma *sensor Fusion* dengan luaran ahir berupa sudut *yaw*, *pitch* dan *Roll*.

A. Perancangan Hardware *IMU*

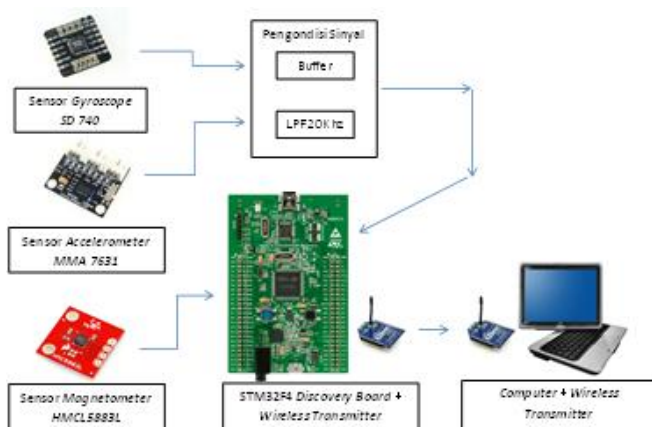
Skema perancangan *Hardware IMU* dan komponen pendukung yang diajukan pada tugas akhir ini dapat diamati pada Gambar 2.

Dalam Skema sistem *IMU* dipilih beberapa sensor yang akan digunakan, untuk accelerometer dipilih MMA7631 sensor ini memiliki keunggulan untuk mendeteksi akselerasi dengan batas *1G* dan *6G*. Sensor ini juga lebih dipilih daripada sensor lain karena memiliki *Bandwidth* mencapai 400Hz dan mengintegrasikan 3 aksis sensor sekaligus.

SD 740 dipilih menjadi sensor gyroscope dalam tugas akhir ini karena sudah mengintegrasikan 3 sensor kecepatan sudut dan terdapat pengondisi sinyal didalamnya diharapkan perubahan bias yang terjadi tidak sebesar sensor yang lain.

HMC5833L dipilih untuk digunakan sebagai sensor magnetometer karena dapat mendeteksi kuat medan magnet 3 aksis secara sekaligus dan satu satunya yang mudah diperoleh dengan harga yang murah di Indonesia.

HMC5883L menggunakan I2C sebagai transfer datanya



Gambar. 2. Skema sistem *IMU* dan komponen Penunjang.

sehingga tidak diperlukan pengondisi sinyal analog, lain halnya dengan SD740 dan MMA7631. SD740 memerlukan sebuah buffer karena impedansi luarannya besar sehingga tanpa buffer maka ada kemungkinan data menjadi tidak *reliable* low pass filter maupun highpass filter tidak dibutuhkan karena menurut datasheetnya sudah terdapat pengondisi sinyal didalamnya. MMA7631 membutuhkan kapasitor yang dipasang pada outputnya untuk menjadi lowpass filter untuk meredam noise yang dihasilkan oleh clock internal dari IC ini, karena resistansi luarannya bernilai 32KOhm maka disarankan oleh datasheet menggunakan kapasitor sebesar 3.3nF untuk Menghasilkan frekuensi *cutoff* sebesar 1507 Hz. Luaran ini kemudian dimasukkan kedalam opamp yang berperan sebagai buffer.

Realisasi perangkat keras dapat diamati pada Gambar 3. Board di bagi kedalam tiga yakni board bawah yang terdiri dari pengondisi sinyal, *accelerometer* dan *gyroscope*. Disertai



Gambar. 3. Realisasi *Hardware* Sistem *IMU*.

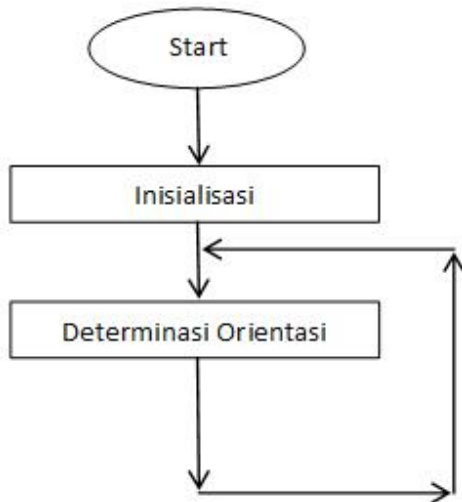
dengan pin pin luaran yakni *Supply* dan konektor untuk membaca tegangan dari alat pengujian yakni gimbal elektronik.

Board kedua adalah STM32F4Discovery yang terletak terdapat dalam dua board tersebut. Board ini merupakan sistem minimum dari mikrokontroler STM32F407VG, disertai pula dengan *debugger Onboard*. STM32F407VG dipilih menjadi pemroses data utama karena sudah menerapkan arsitektur ARM 32 Bit dan mampu beroperasi pada kecepatan clock mencapai 168 Mhz

Board terakhir adalah board yang berisi *Magnetometer* dan *Xbee Wireless Transmitter*, kedua komponen ini dirancang diletakkan diatas karena dalam rancangan awal peletakan pada board dibawah menyebabkan pembacaan magnetometer terganggu dan pengiriman data Xbee menjadi sangat tidak lancar, oleh karena itu board sensor dirancang ulang menjadi 3 board seperti pada Gambar 3.

B. Perancangan Software IMU

Flowchart umum program dapat diamati pada Gambar 4. Diagram Blok Determinasi Orientasi dapat diamati pada Gambar 5.



Gambar. 4. Flowchart umum program.

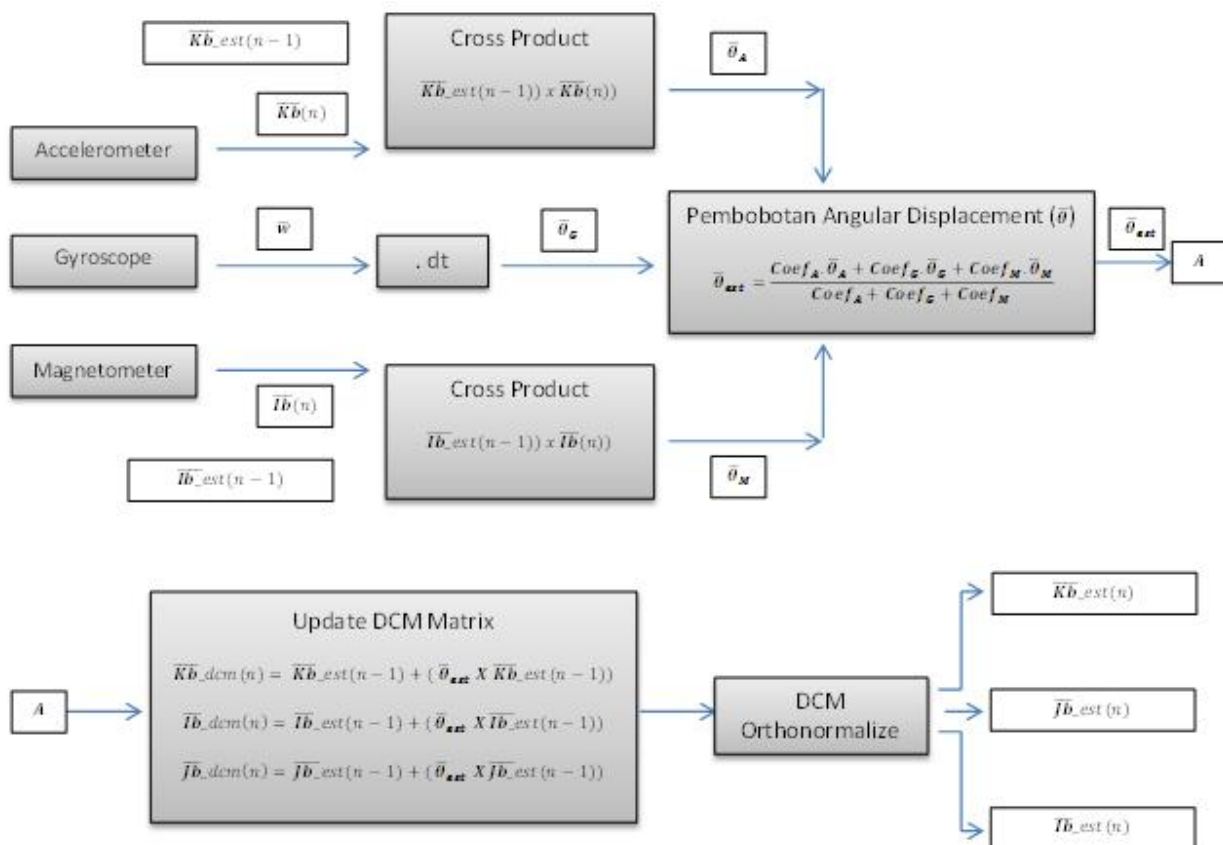
Fase inisialisasi adalah fase untuk mengawali kinerja mikrokontroller seperti pengaturan *register register ADC, Timer, GPIO* dan *I2C* serta memberi nilai awal pada beberapa

variabel seperti matriks DCM awal.

Kemudian dilanjutkan dengan program determinasi orientasi yang menerima masukan berupa data dari *Accelerometer* berupa vektor *Zenit(Kb)*, *gyroscope* berupa vektor kecepatan sudut (ω) kemudian *magnetometer* berupa vektor Utara (*Ib*).

Vektor kecepatan kemudian dikalikan dengan periode *update rate flowchart* program (dt) sehingga menjadi sebuah nilai simpangan sudut atau *gyroscope angular displacement* (θ_g). Data dari *accelerometer* dan *magnetometer* kemudian dapat diolah menjadi *angular displacement* juga sesuai dengan rumus pada blok *cross product* pada gambar 5. 3 jenis *angular displacement* ini lah yang kemudian dijadikan satu dengan memberikan bobot pada masing masing dengan rumus sesuai pada blok pembobotan pada Gambar 5. Hasil penyatuan ini yang kemudian dipakai untuk mencari estimasi sesungguhnya dari vektor gravitasi, vektor utara dan vektor barat. Ketiga vektor tersebut apabila dirangkai dalam bentuk matriks 3x3 maka dapat disebut dengan *Direction Cosine Matrice (DCM)*. *DCM* ini lantas kemudian tidak dapat langsung di gunakan sebagai mestinya karena ada kemungkinan muncul error sehingga vektor Masing masing tidak saling orthogonal kembali. Blok *DCM Orthonormalize* inilah yang bertujuan untuk mengembalikan ke orthogonal dari *DCM*. Setelah *DCM* di ortogonalnkan maka langkah selanjutnya adalah menggunakan nilai nilai tersebut untuk digunakan sebagai inisialisasi pada *looping* program Berikutnya.

Untuk aplikasi kontrol, luaran *DCM* ini kemudian diolah kembali untuk menghasilkan sudut *pitch yaw* dan *roll*.



Gambar. 5. Diagram Blok *Determinasi Orientasi*.

Serangkain program diatas di masukkan kedalam mikrokontroller menggunakan program Bawaan STM32F4 yakni Keil luaran ARM.

III. PENGUJIAN DAN ANALISIS

Pada bab ini akan dipaparkan hasil pengujian *IMU* beserta analisisnya. Pengujian dilakukan dengan menggunakan gimbal elektronik seperti pada Gambar 6. Dan *quadrotor* namun tidak diterbangkan.



Gambar 6. Gimbal elektronik dalam pengujian sudut estimasi yaw.

A. Analisis Performa Statis dan Dinamis

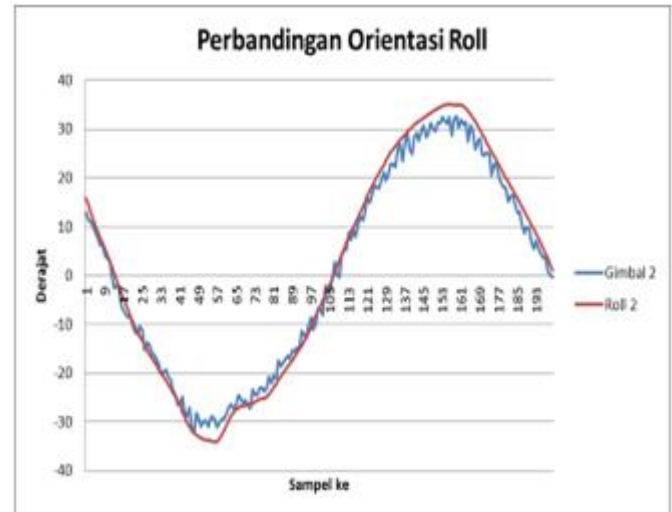
Pengujian dilakukan dengan merata rata 30 sampel data dimana setiap sampel data merupakan perhitungan *RMS error* dari 200 titik *sampling* data. Masing masing 5 sampel dari 30 dilakukan dengan frekuensi *update* data yang berbeda beda. Bobot masing masing sensor adalah tetap setiap pengujian. Referensi untuk perhitungan *RMS Error* Statis didapat dengan mengamati busur pada gimbal, Referensi pada *RMS Error* dinamis didapat dengan mengamati tegangan luaran *potensiometer* dan mengkonversinya ke bentuk sudut. Hasil pengujian di tampilkan pada Tabel 1.

Tabel 1.
Performa *RMS Error* Statis dan Dinamis

Kondisi Pengujian	<i>RMS Error Pitch</i>	<i>RMS Error Roll</i>	<i>RMS Error Yaw</i>
Dinamis	3.335	2.203	2.637
Statis	1.949	0.240	0.484

Dari Tabel 1. Dapat diamati bahwa error statis lebih kecil dari pada error dinamis, hal ini dapat disebabkan karena penggunaan algoritma *sensor fusion* yang bergantung pada pendekatan matematis sederhana yang akan bernilai kecil saat *dt* bernilai kecil atau mendekati nol, hal ini perlu dibuktikan dengan membandingkan *RMS error* terukur terhadap frekuensi *update rate*. Pengujian dapat diamati pada bab selanjutnya. Kemungkinan kedua yang menyebabkan error dinamis lebih

besar adalah metode pengujiannya yang menggunakan gimbal elektronik yang tersusun atas *potensiometer* yang notabnya mengandung *noise* semisal dari gesekan mekanik Gimbal. Hal ini didukung dengan mengamati sampel hasil grafik pengujian *RMS Error* dinamis pada Gambar 7.



Gambar 7. Perbandingan Orientasi Roll terukur, Gimbal VS IMU.

B. Analisis Pengaruh Frekuensi Update Data terhadap Performa IMU

Pengujian dilakukan dengan mengamati perubahan *RMS Error* terhadap *Frekuensi Update* data. Kelompok data yang digunakan adalah sama dengan kelompok data pada pengujian di Bab Analisa performa statis dan dinamis. Hasil pengujian dapat diamati pada Tabel 2.

Tabel 2.
Performa *RMS Error* Terhadap Frekuensi

Frekuensi Update Data	<i>RMS Error Pitch</i>	<i>RMS Error Roll</i>	<i>RMS Error Yaw</i>
50Hz	2.722	1.264	2.586
100Hz	2.751	1.434	1.833
150Hz	3.023	1.267	1.508
200Hz	2.399	1.011	1.189
250Hz	2.299	1.148	1.113
300Hz	2.656	1.205	1.134

Dari Tabel 2. Dapat diamati bahwa tampak terjadi perubahan *RMS Error* namun tidak memiliki pola pasti pada *RMS Error* sudut *Pitch* dan *Roll*. Berbeda dengan Hasil pengujian pada sudut *Yaw* yang menunjukkan adanya penurunan error terhadap naiknya frekuensi *update* data. Hasil analisa mengenai tidak tampaknya perubahan untuk sudut *Pitch* dan *Roll* adalah bisa disebabkan oleh frekuensi yang dipilih merupakan sudah frekuensi yang stagnan terhadap perubahan Error. Data dari sudut *yaw* mewakili bahwa frekuensi mempengaruhi besarnya *RMS Error*.

C. Analisis Pengaruh Perubahan bobot Sensor Terhadap Performa IMU

Pengujian dilakukan dengan melakukan pengambilan data dengan perubahan bobot sebagai variabel. Frekuensi *update* data dipilih 100Hz. Untuk sudut *Pitch* dan *roll* dilakukan perata rata 20 sampel data *RMS Error*, dimana 5 sampel adalah sudut statis 0 derajat, 5 sampel adalah statis di 45 derajat, 5 sampel statis di -45 derajat dan 5 sampel merupakan data dinamis. Sedangkan untuk yaw ditambahkan 5 sampel statis di 135 derajat, 5 sampel statis di -135 derajat dan 10 sampel data dinamis. Setiap sampel adalah *RMS Error* dari 200 titik data. Hasil pengujian dapat diamati pada Tabel 3. G adalah Bobot sensor *Gyroscope*, A adalah bobot sensor *Accelerometer* dan M adalah bobot dari sensor *Magnetometer*.

Tabel 3.
Performa *RMS Error* Terhadap Perubahan Bobot

Kondisi Pengujian	<i>RMS Error Pitch</i>	<i>RMS Error Roll</i>	<i>RMS Error Yaw</i>
G=1; A=0; M=0	24.244	27.029	57.326
G=1; A=0.1; M=0.1	3.336	2.354	2.938
G=1; A=0.5; M=0.5	3.386	2.268	2.881
G=1; A=1; M=1	3.038	2.463	2.307

Perubahan Bobot *Accelerometer* dan *Magnetometer* secara tidak langsung merubah Pengaruh bobot *Gyroscope* kearah yang berlawanan oleh karena itu bobot *gyroscope* dibuat tetap, walau sebenarnya pengaruhnya melemah.

Semakin rendah bobot *Accelerometer* dan *Magnetometer* semakin besar *RMS Error* yang dihasilkan Hal ini di akibatkan oleh semakin lemahnya pengaruh *Accelerometer* dan *Magnetometer* yang berfungsi sebagai penghilang *Drift* yang dihasilkan oleh Penggunaan *Gyroscope*. Namun bukan berarti dengan meningkatkan terus nilai bobot *Accelerometer* dan *Magnetometer* dapat membuat *RMS error* menjadi lebih kecil, Dengan menambah bobot *Accelerometer* berarti kemungkinan *noise* akibat getaran dari *accelerometer* masuk mengganggu determinasi orientasi. Hal ini dibuktikan dengan mengamati dua buah sampel dengan konstanta *Accelerometer* yang berbeda jauh, Perbandingan dapat diamati pada Gambar 8. Pada Gambar 8.a dapat dilihat bahwa estimasi luaran *accelerometer* tampak tidak halus dibandingkan dengan data luaran estimasi pada Gambar 8.b hal ini membuktikan hipotesa awal bahwa penambahan bobot *accelerometer* membuat *IMU* rentan terhadap *noise*.

D. Analisis Performa IMU pada Quadrotor

Pengujian dilakukan dengan cara menempatkan *IMU* diatas *quadrotor* yang di hidupkan namun tidak diterbangkan dan dimiringkan -9 derajat. Pengujian yang dilakukan adalah pengujian statis dengan sudut *pitch* sebagai luarannya. Diambil beberapa Data dengan merubah konstanta bobot sensor dan diamati pengaruhnya. Total Telah diambil tiga buah data



(a)



(b)

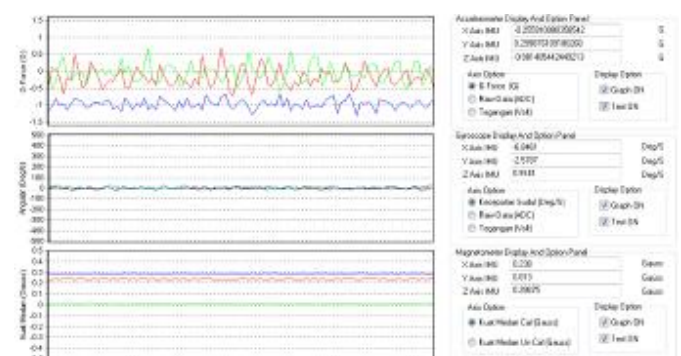
Gambar 8. Perbandingan Pengujian *RMS Error Roll* dengan pembobotan sensor yang berbeda (a) Bobot Sensor adalah G = 1; A = 1 ; M = 1 ; (b) Bobot sensor adalah G = 1; A = 0.1 ; M = 0.1 ;

dengan merubah koefisien *accelerometer* saja. Hasil pengujian dapat diamati pada Tabel 4.

Tabel 4.
Performa *RMS Error* Diuji di Quadrotor

Kondisi Pengujian	<i>RMS Error Pitch</i>
G=1; A=0.1; M=0.1	4.343
G=1; A=0.04; M=0.04	4.457
G=1; A=0.01; M=0.01	4.035

Dari Tabel 4 dapat di tarik keimpulan bahwa dengan memperkecil bobot *accelerometer* akan diperoleh *RMS Error*



Gambar 9. Kondisi *Realtime Sensor* saat *Quadrotor* Hidup.

yang kecil. Hal ini disebabkan noise di *accelerometer* akibat getaran *quadrotor* adalah sangat tinggi sesuai dengan grafik pada Gambar 9. Sehingga dengan memperkecil bobot *accelerometer* maka *noise* akan berkurang.

Informasi penting lainnya yang dapat di ambil dari mengamati Gambar 9. Adalah ketahanan Gyroscope dan Magnetometer terhadap Vibrasi dari Quadrotor jauh lebih baik dari pada Accelerometer (grafik paling atas).

Perlu di rancang suatu algoritma tambahan untuk mengatasi kondisi pakai pada quadrotor ini, karena jika bobot accelerometer terus di kecilkan maka drift akan mendominasi keseluruhan hasil determinasi orientasi. Perlu di lakukan pengujian RMS Error Dinamis agar kehandalan dapat diuji sepenuhnya.

IV. KESIMPULAN

Inertial Measurement Unit dengan Algoritma yang diajukan dalam Tugas Ahir ini dapat direalisasikan dengan *RMS Error* Statis mencapai < 2 Derajat dan *RMS Error* Dinamis Mencapai < 3.4 Derajat.

Pengujian Performa terhadap laju *Update* data dilakukan mulai dari frekuensi 50Hz sampai dengan 300 Hz. Tidak menutup kemungkinan untuk diperasikan dengan frekuensi yang lebih tinggi.

Frekuensi update Data berpengaruh terhadap *RMS Error* yang akan dihasilkan, Semakin tinggi frekuensi semakin kecil *RMS Error*. Membesarkan bobot *Accelerometer* akan memperkecil pengaruh drift dari penggunaan *Gyroscope* namun juga akan menambah *noise* dari *accelerometer* itu sendiri.

DAFTAR PUSTAKA

- [1] Madwick S, Harrison A, Estimation of IMU and MARG orientation using a gradient descent algorithm. Proceedings of the IEEE International Conference on Rehabilitation Robotics, Zurich Science City, Switzerland (2011).
- [2] Sarlino Electronics, "DCM Tutorial" < URL : http://www.starlino.com/dcm_tutorial.html> (2011).